HTTP

Michael Gfeller

Folien: Silvan Gehrig

HSR
HOCHSCHULE FÜR TECHNIK
RAPPERSWIL

FHO Fachhochschule Ostschweiz

Lernziele

Die Vorlesung soll die Teilnehmer befähigen, HTTP Request und Response im Grundsatz und mit deren wesentlichen Eigenschaften zu verstehen und dessen Konzepte fachgerecht einzusetzen.

Die Teilnehmer...

- **verstehen, wie ein Web-Request zustande kommt und kennen dessen Ablauf.**
- **können HTTP Protokoll-Probleme zwischen Client und Server interpretieren und deren Fehlerquelle einschätzen.**
- **kennen einige der wichtigsten HTTP Header mit deren Funktionalitäten.**

Table of Contents

HYPERTEXT TRANSFER PROTOCOL

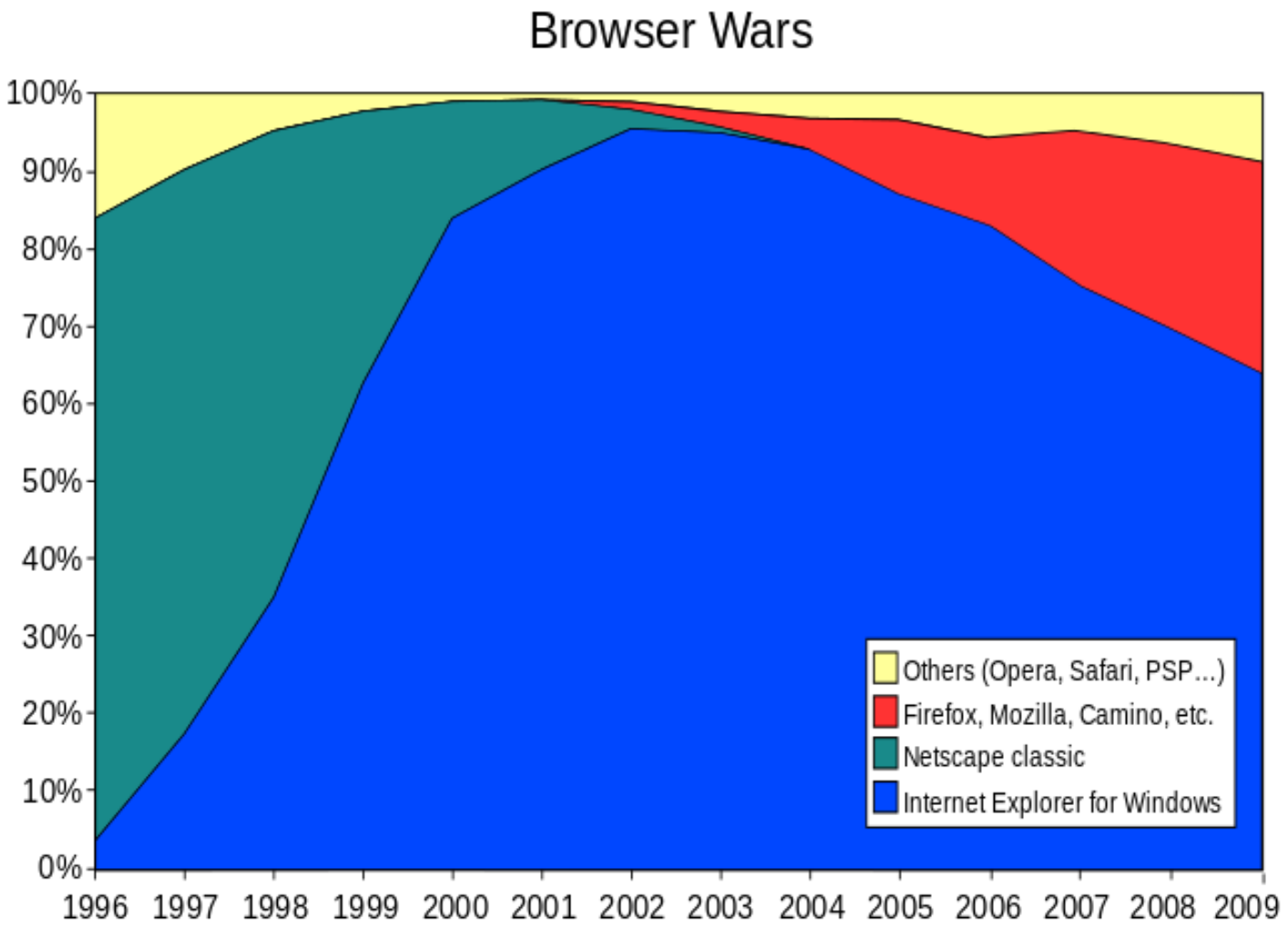
INTRODUCTION

HTTP

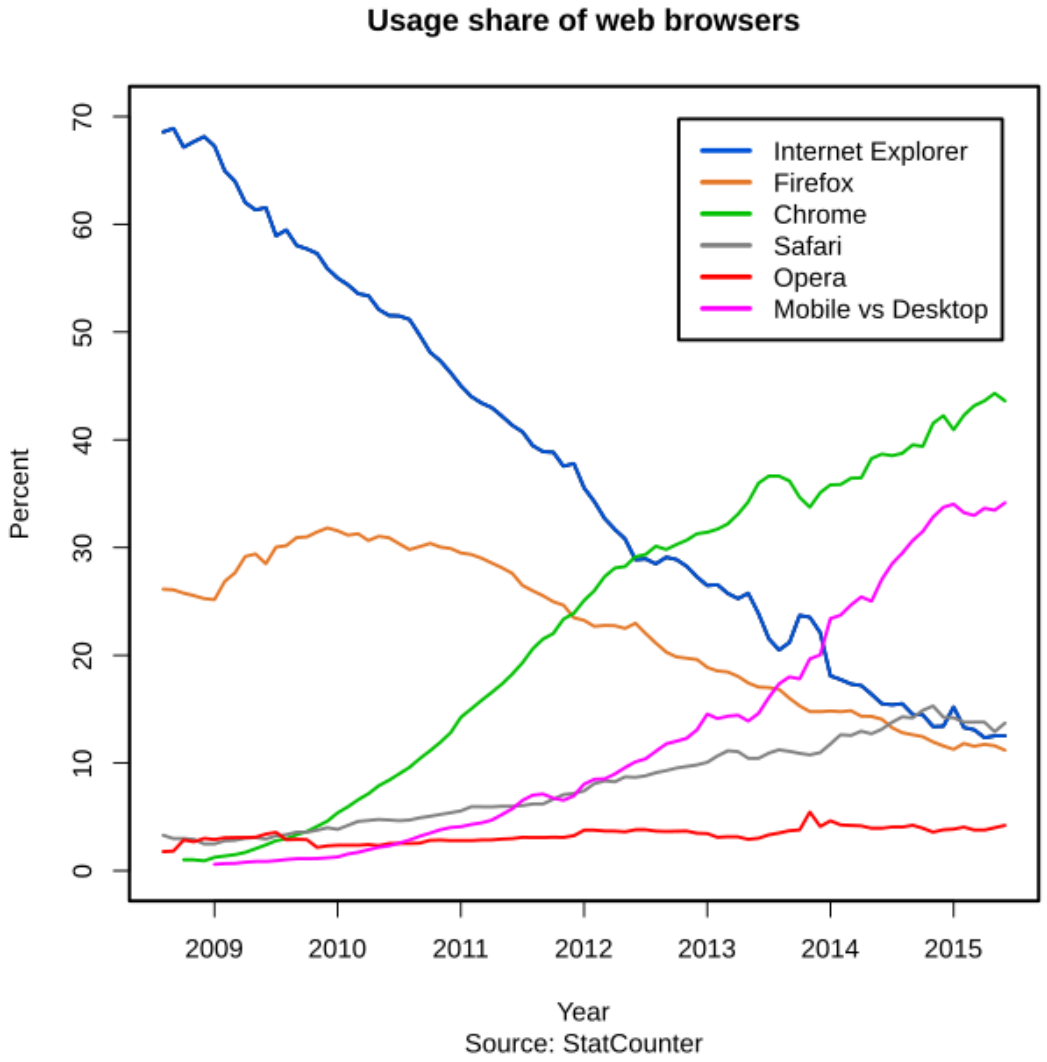
HTTP History

Year	Comment	
1962	ArpaNet	
as of 1981	TCP / IP as network protocol (→ Internet) Layer 7 Protocols: POP, SMTP, FTP, News etc.	
as of 1989	Development of HTTP at CERN Tim Berners-Lee	
1991	HTTP/0.9 Proprietary implementation Netscape, IE	
1996	HTTP/1.0 as RFC1945 der IETF	*First Browser War*
1999	HTTP/1.1 as RFC 2616 / 2617 der IETF	
2008		*Second Browser War*
2014	HTTP/1.1 as RFC 7230-7235	
ab 2012	SPDY as predecessor of HTTP/2.0	
ab 2015	HTTP/2.0 as RFC 7540	

First Browser War 1996-2009



Second Browser War 2009-2015



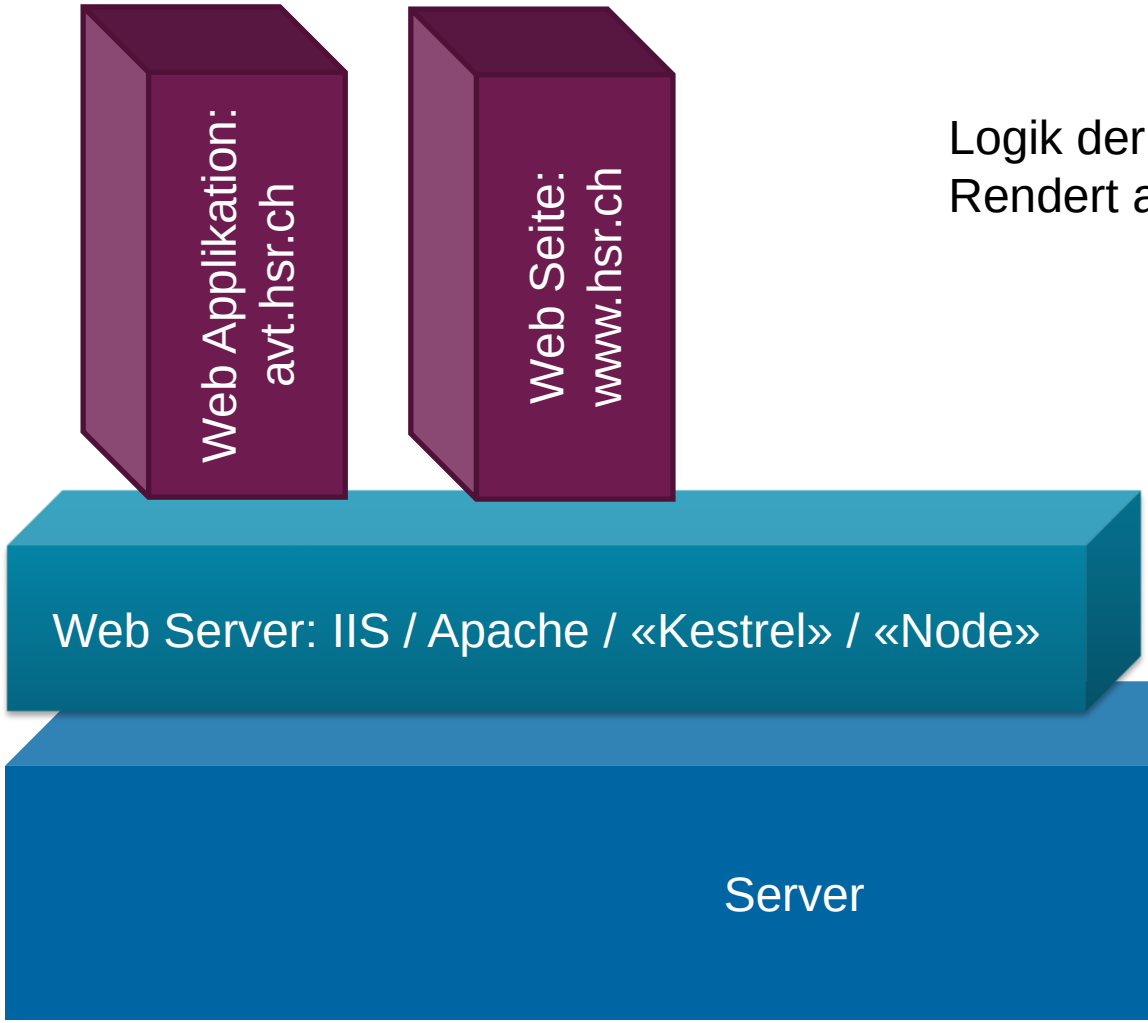
http://gs.statcounter.com/

DEMO INTRO

Client / Server Architecture

- **Client**
  - Workstation / PCs / …
  - Rely on server resources
- **Server**
  - (Powerful) Computers
  - Dedicated to mangage shared resources

Web Server und Web Applikation oder Web Seite



Logik der Webseite:
Rendert anhand der URL die Antwort

Nimmt Anfragen vom Netzwerk entgegen und übergibt diese an die jeweilige Web Applikation.

Betriebssystem & Hardware

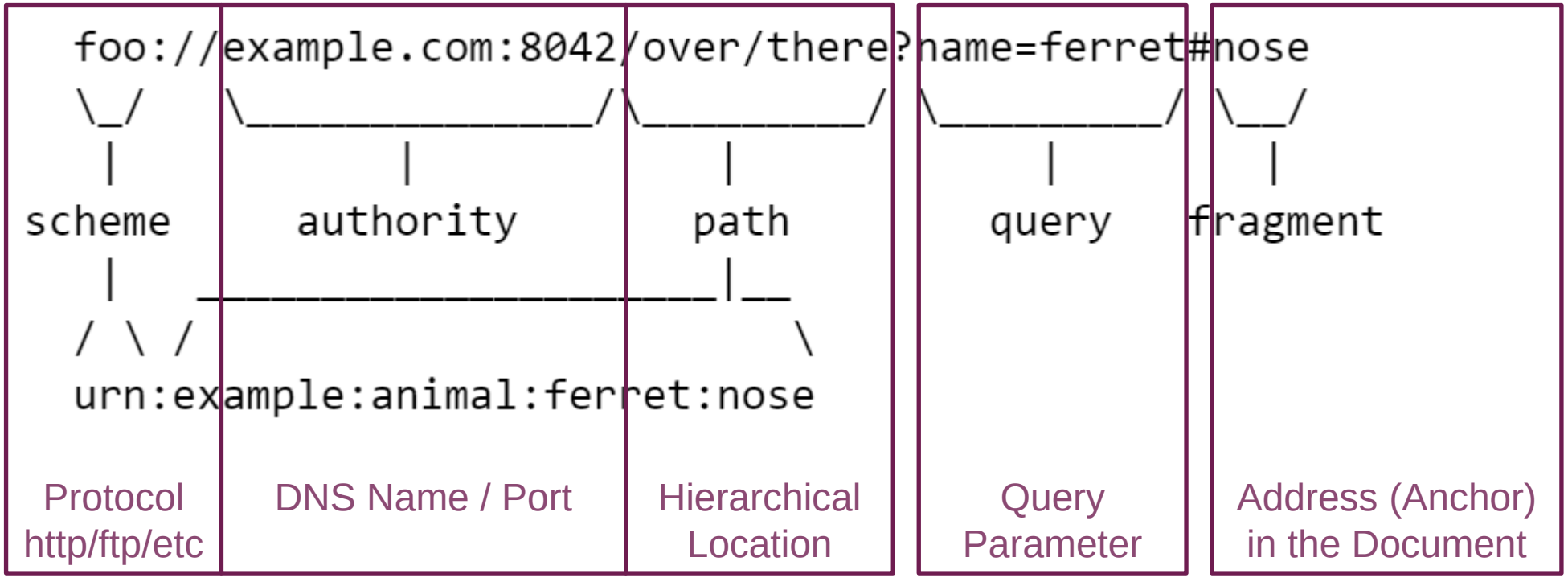
URI SCHEMA

HTTP

URL Schema

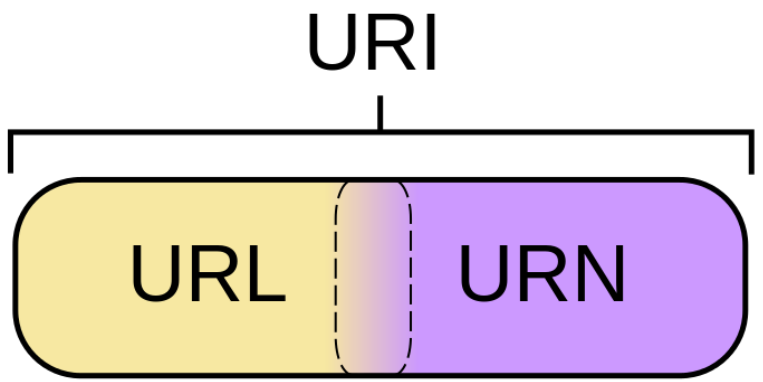
- **<scheme name> : <hierarchical part> [? <query>] [# <fragment>]**

- **Example:**



HTTP URI

- **URI = Unified Resource Identifier (URI)**
  - is a string of characters used to identify a name of a resource
  - URL and URN are URIs
- **URL = Unified Resource Locator (URL)**
  - is a reference to a resource that specifies the location of the resource on a computer network and a mechanism for retrieving it
  - URLs may contain a URN
  - Limited schema range
- **URN = Unified Resource Name (URN)**
  - is a string of characters used to identify a name of a resource.
  - Independent from location / persistence / …
  - Limited to schema “URN”



HTTP URI: Beispiele

URI	URN
tel:+1-816-555-1212	+1-816-555-1212
https://www.book-shop.demo/9780596517748	9780596517748
https://www.hsr.ch/Images/logo_hsr.svg	/Images/logo_hsr.svg
telnet://192.0.2.16:80/	192.0.2.16:80

URI Encoding

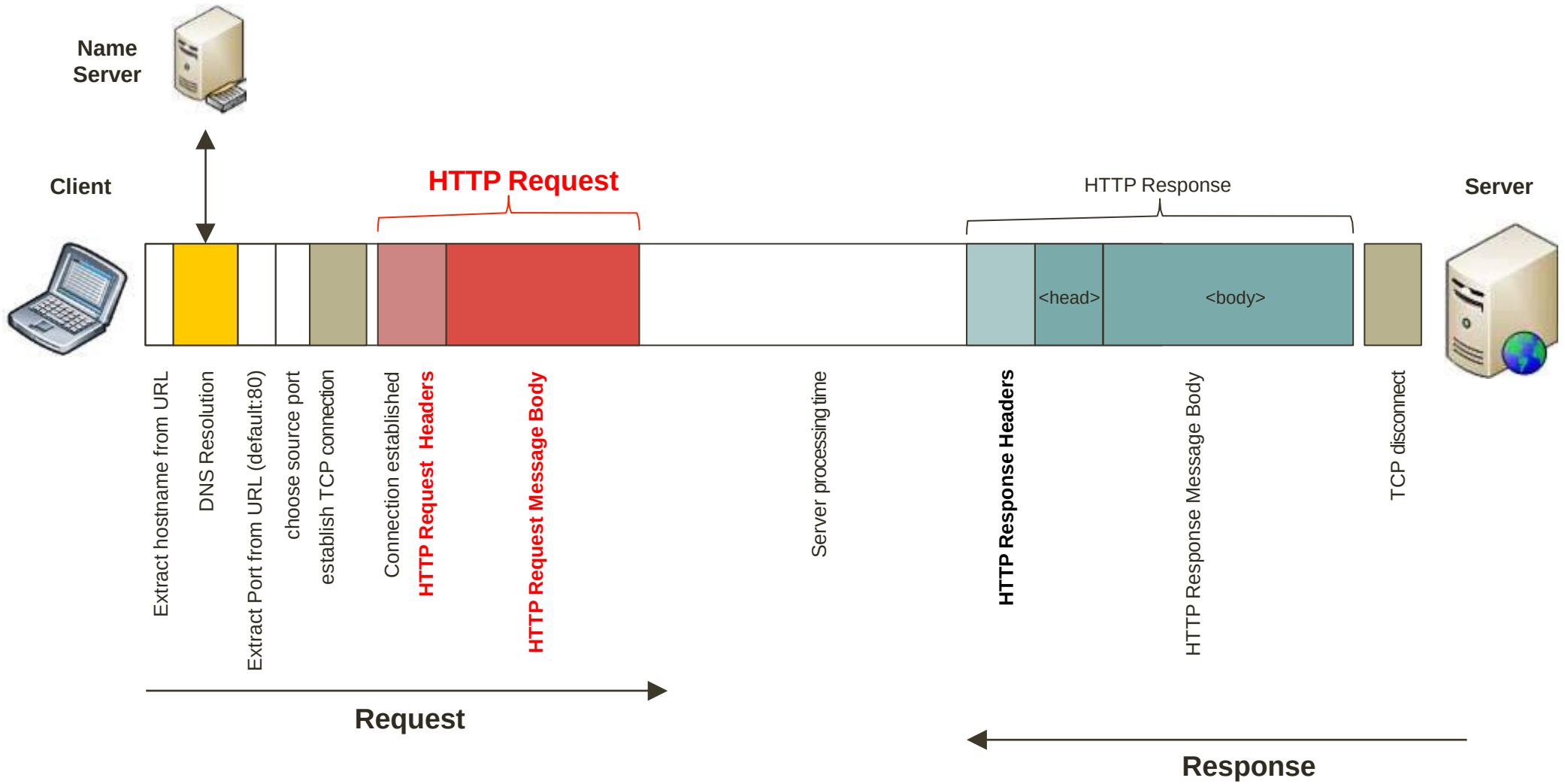
- **Special Characters must be encoded**
- **Important for control characters**

- **Example:**
  - ‘/’ must be converted %2F
  - ‘ ‘ (space) %20
  - ‘&’ %26
  - ‘#’ %23
  - ‘?’ %3F

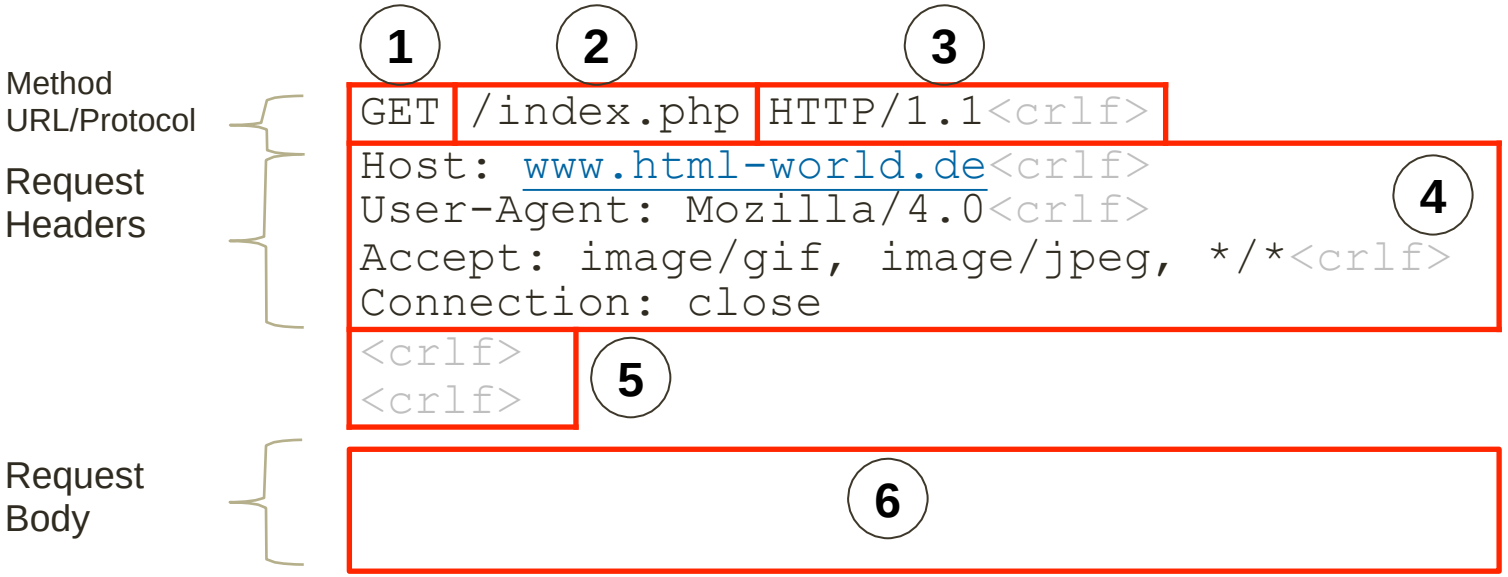
REQUEST

HTTP

Establish an HTTP Connection



HTTP Request



1. Method
2. Address (URL Path)
3. Protocol Version (HTTP/0.9 | HTTP/1.0 | HTTP/1.1. | HTTP/2.0)
4. Request Headers (optional, depends on the application)
5. Header/Body Separator (2 x crlf)
6. Request Body (used for example in <form> Data Transmitions)

HTTP Methods

Method	Description
GET	The GET method is used to **retrieve information** from the given server using a given URI. Requests using GET should only retrieve data and should have **no other effect on the data**.
HEAD	Same as GET, but transfers the **status line and header section** only.
POST	A POST request is used to **send data to the server**, for example, **new** customer information, *file upload*, etc. using *HTML forms*.
PUT	**Replaces all current representations** of the target resource with the uploaded content.
DELETE	**Removes all current representations** of the target resource given by a URI.
CONNECT	Establishes a tunnel to the server identified by a given URI.
OPTION	Describes the communication options for the target resource.
TRACE	Performs a message loop-back test along the path to the target resource.

HTTP GET vs POST Methods

- **GET data is placed into the URL (as query string)**

```
GET /search?query=abc HTTP/1.1<crlf>
Host: www.html-world.de<crlf>
User-Agent: Mozilla/4.0<crlf>
Accept: image/gif, image/jpeg, */*<crlf>
Connection: close
<crlf>
<crlf>
```

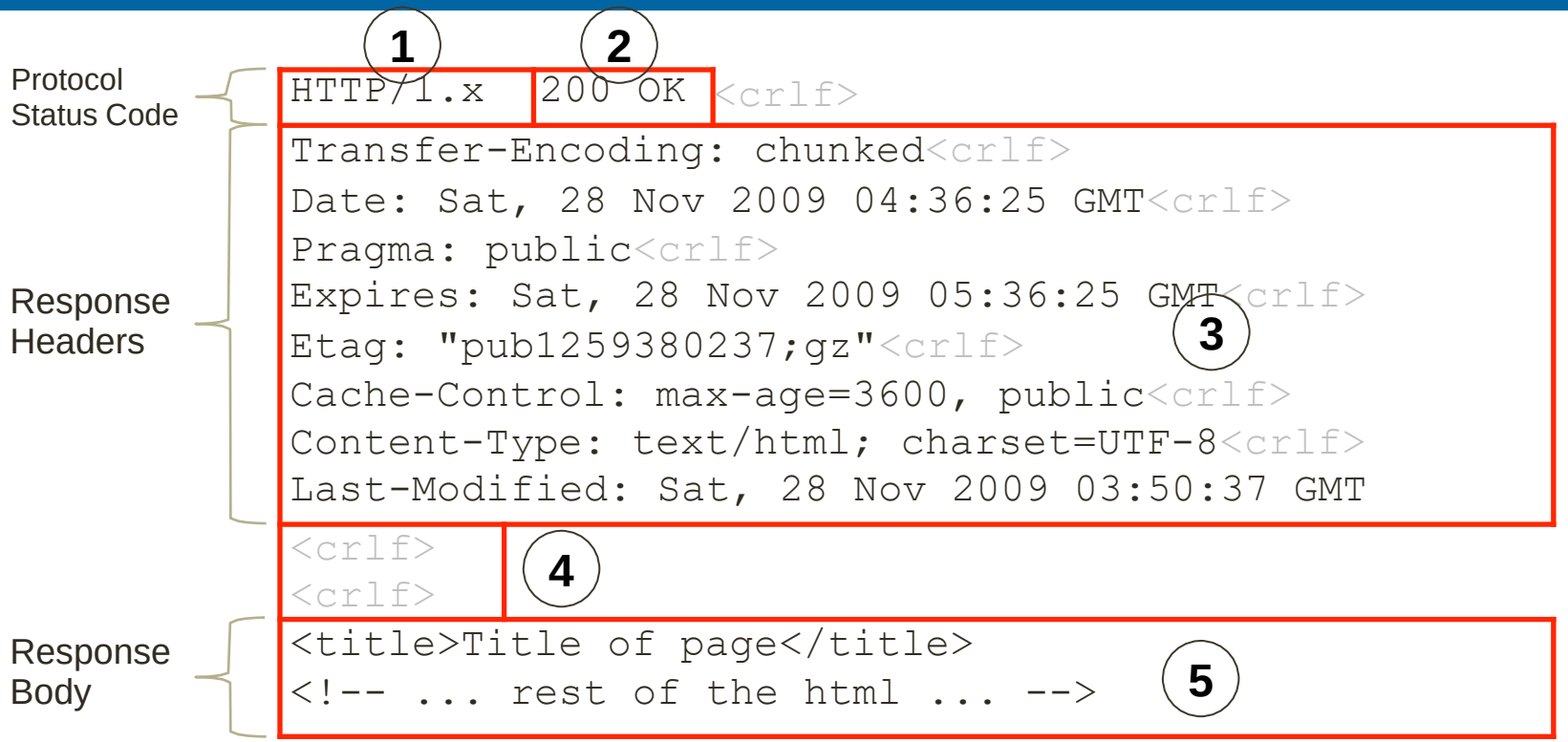
- **POST data is placed into the request body**

```
POST /search HTTP/1.1<crlf>
Host: www.html-world.de<crlf>
User-Agent: Mozilla/4.0<crlf>
Accept: image/gif, image/jpeg, */*<crlf>
Connection: close
<crlf>
<crlf>
query=abc
```

RESPONSE

HTTP

HTTP Response



1. Protocol Version (HTTP/0.9 | HTTP/1.0 | HTTP/1.1. | HTTP/2.0)
2. Status Code
3. Response Headers (optional, depends on application)
4. Header/Body Separator (2 x crlf)
5. Response Body (The returning HTML Code)

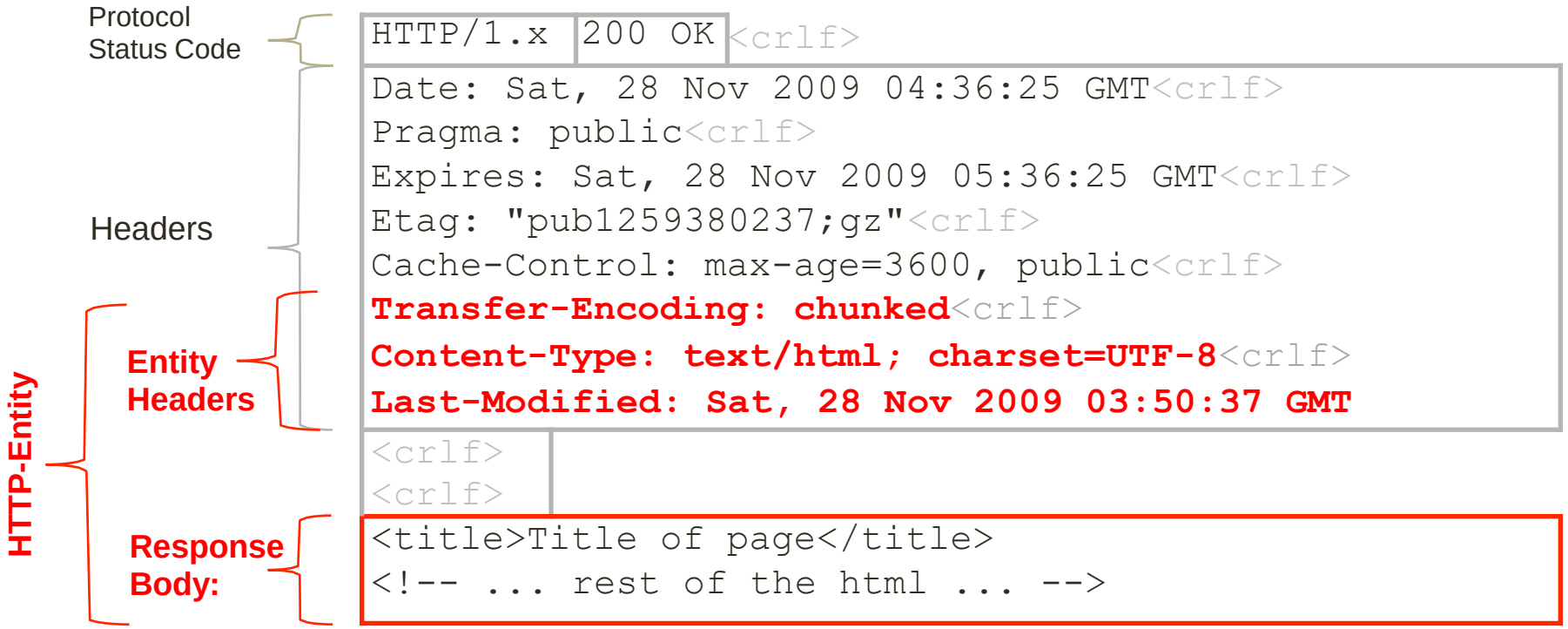
HTTP Response Status Codes

CODE	
1xx	Informational
2xx	Successful 200 OK 201 Created 204 No Content
3xx	Redirection 301 Moved Permanently
4xx	Client Error 400 Bad Request 401 Unauthorized 403 Forbidden 404 Not Found
5xx	Server Error 500 Internal Server Error 505 HTTP Version Not Supported
9xx	Non-Standard Codes

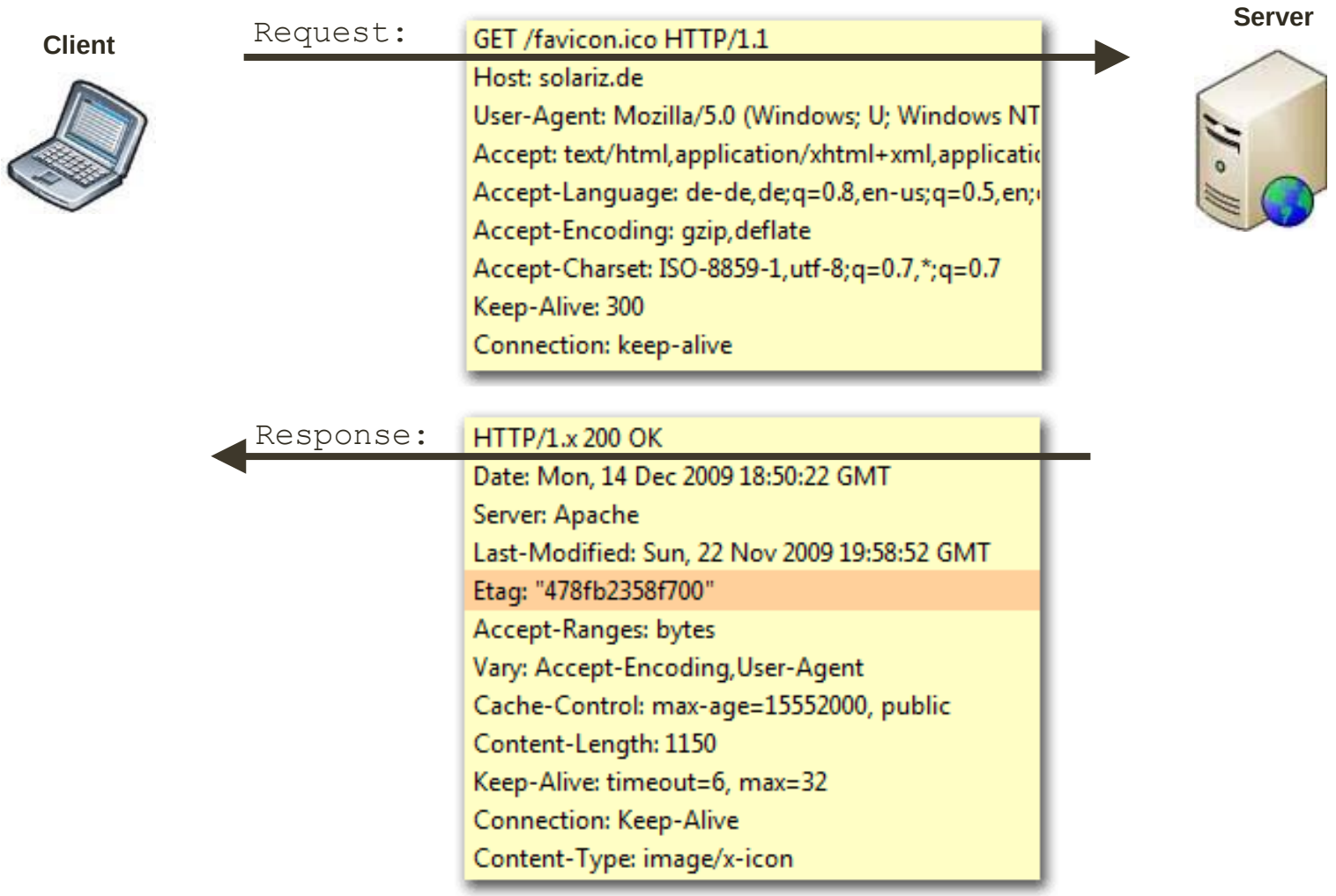
HEADERS

HTTP

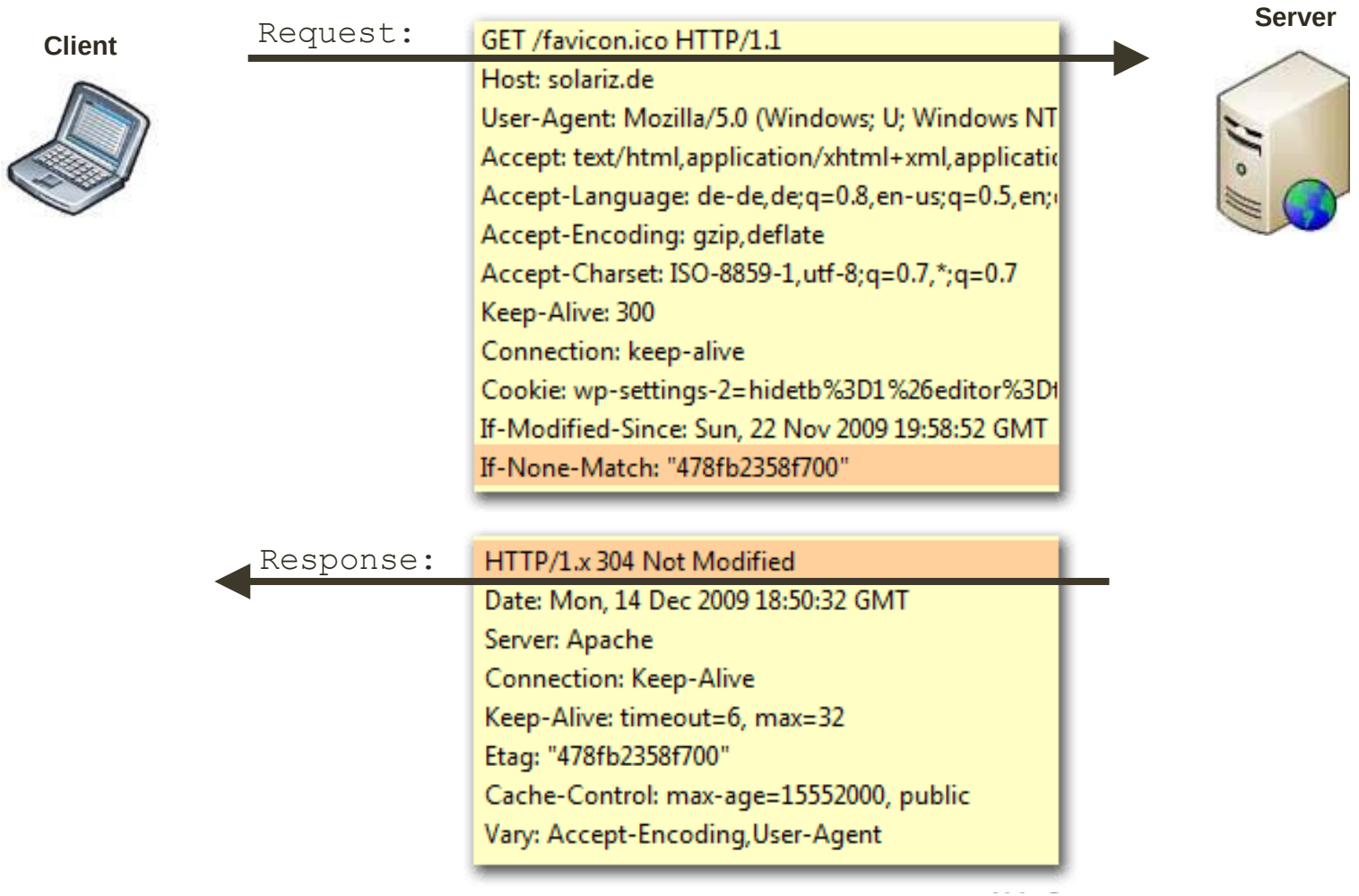
Entity (Response)



Response handling with Etag 1/2



Response handling with Etag 2/2



Headers Overview (Request & Response)

Request

Request Headers

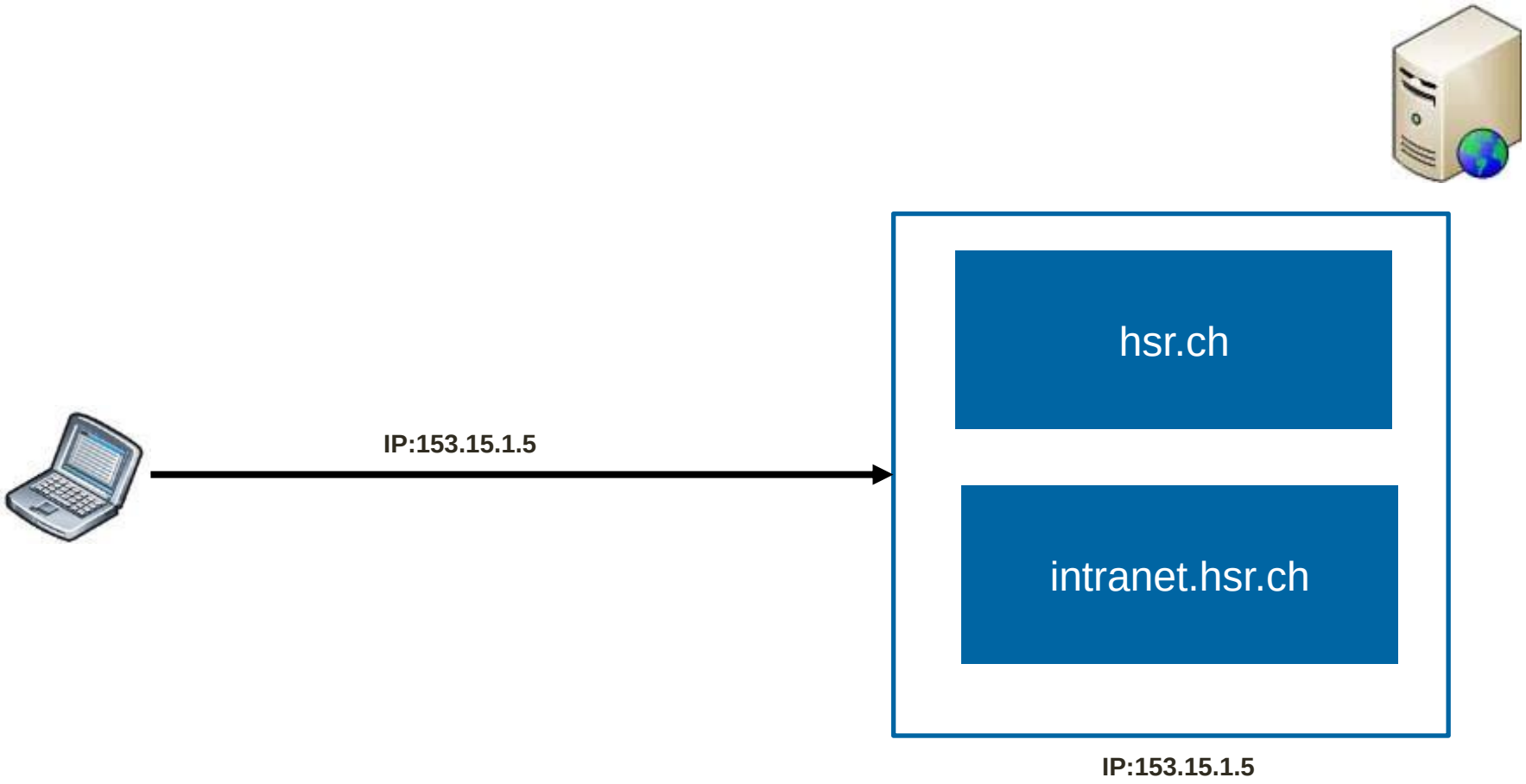
```
GET /index.php HTTP/1.1<crlf>
Host: www.html-world.de<crlf>
User-Agent: Mozilla/4.0<crlf>
Accept: image/gif, image/jpeg, */*<crlf>
Connection: close
<crlf>
<crlf>
```

Response

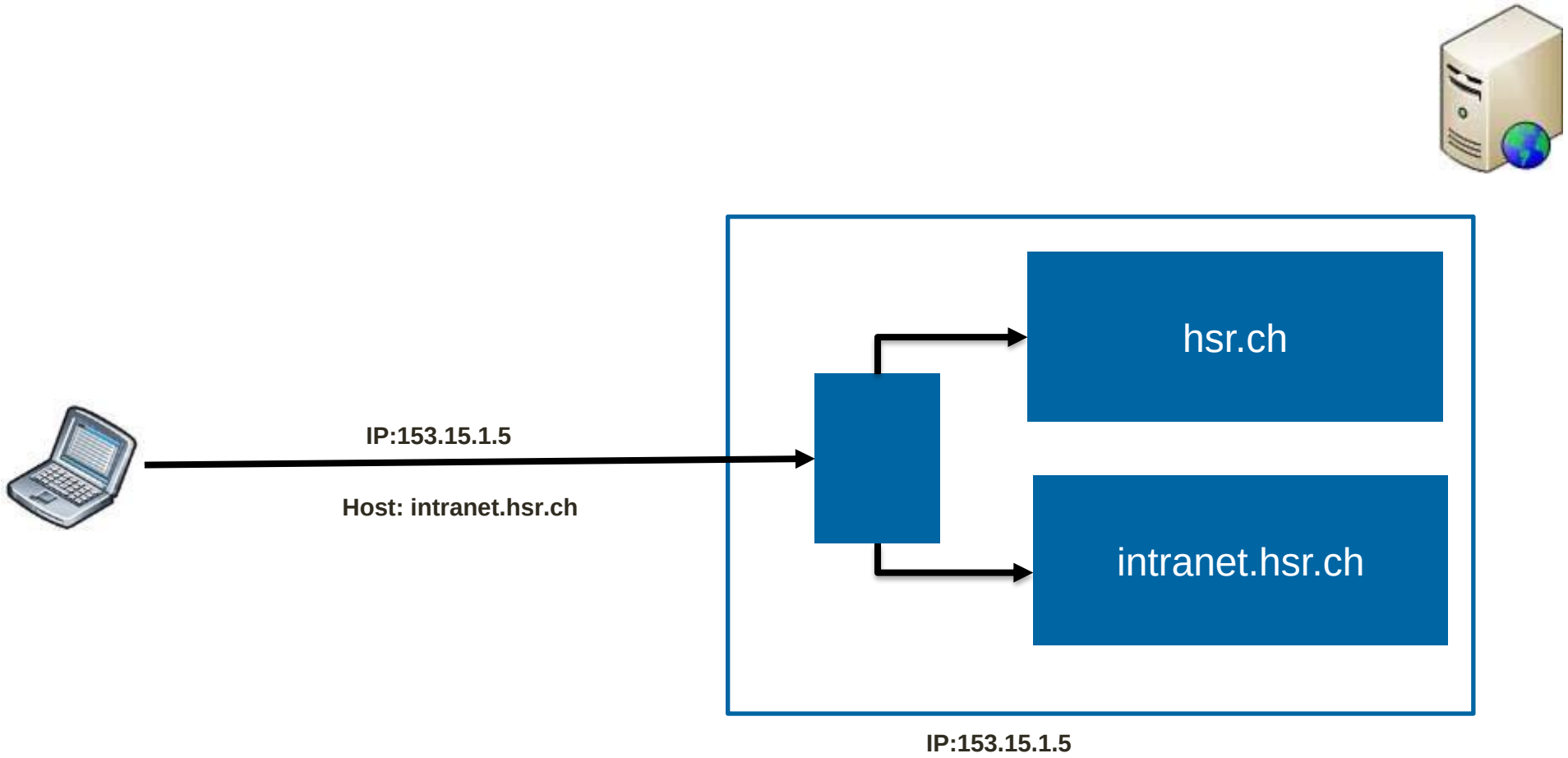
Response Headers

```
HTTP/1.x 200 OK<crlf>
Transfer-Encoding: chunked<crlf>
Date: Sat, 28 Nov 2009 04:36:25 GMT<crlf>
Pragma: public<crlf>
Expires: Sat, 28 Nov 2009 05:36:25 GMT<crlf>
Etag: "pub1259380237;gz"<crlf>
Cache-Control: max-age=3600, public<crlf>
Content-Type: text/html; charset=UTF-8<crlf>
Last-Modified: Sat, 28 Nov 2009 03:50:37 GMT
<crlf>
<crlf>
<title>Title of page</title>
<!-- ... rest of the html ... --
>
```

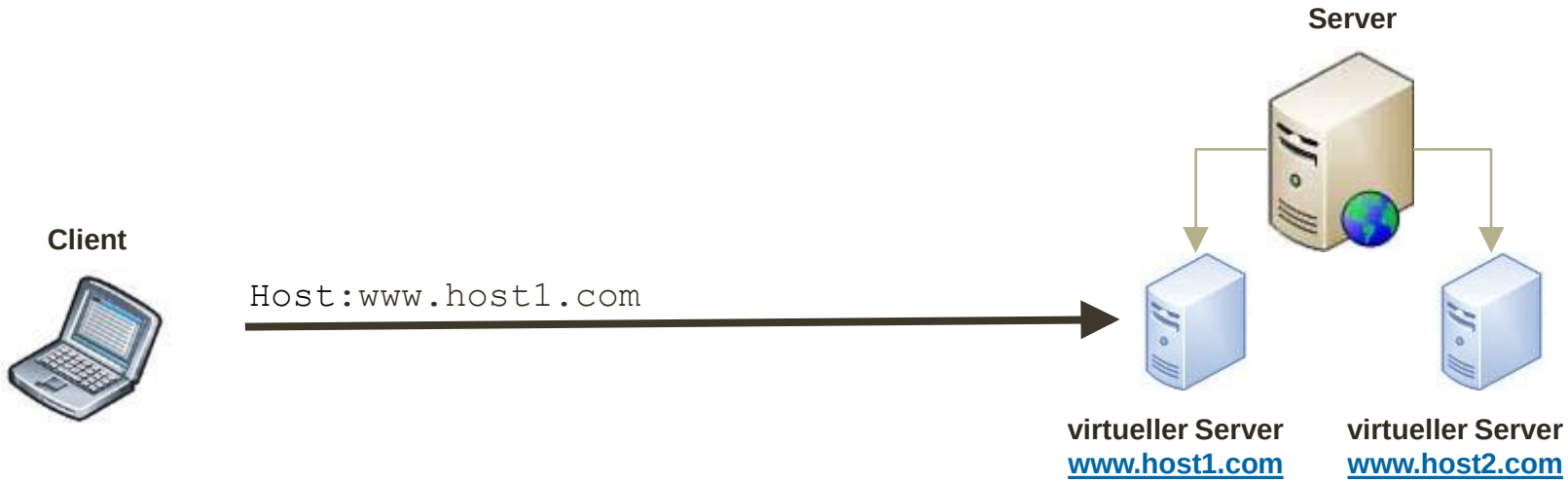
Host Request Header



Host Request Header

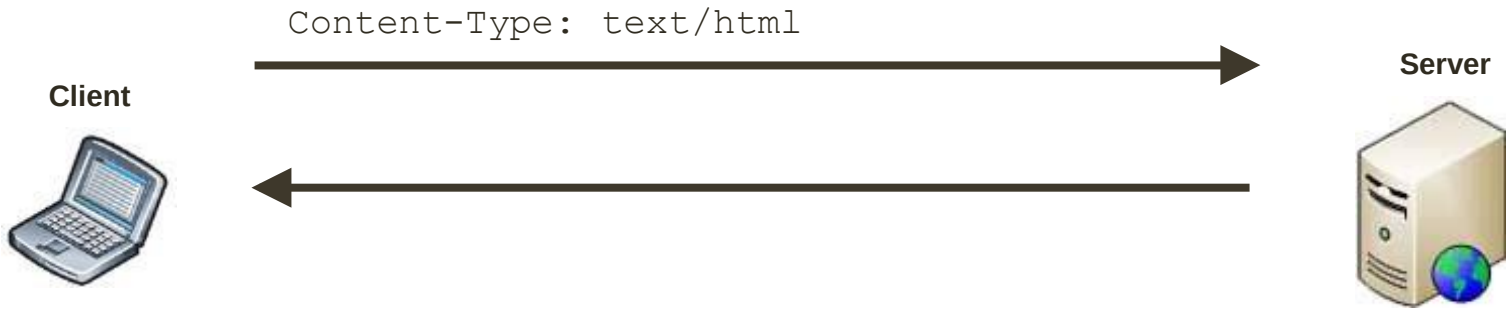


Host Request Header



- **The same IP address can be used for multiple hosts**
- **To determine the virtual server the host name is by-passed**

Content-Type Entity Header

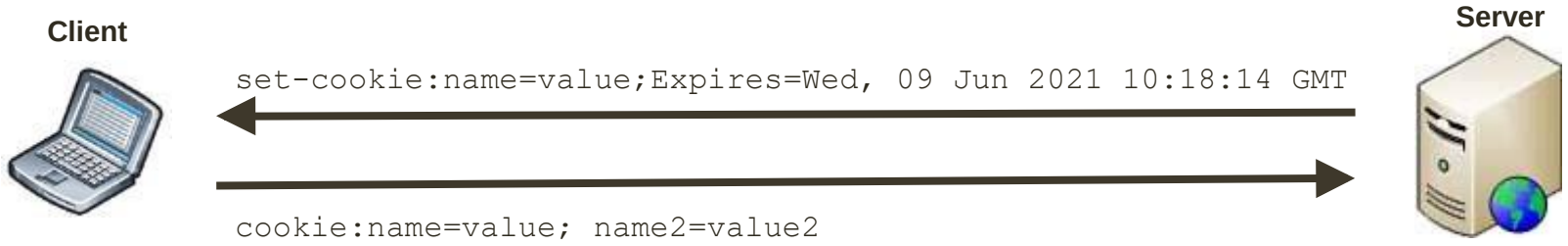


- **Declares the parsing / interpreting format for the Client/Server**
- **The Content-Type is declared in MIME-Type format**
  - Specifies the media type (text, video, audio, application, …) and a subtype (html)

MIME-Types

Type	Description	Example of typical subtypes
text	Represents any document that contains text and is theoretically human readable	text/plain text/html text/css ~~text/javascript~~
image	Represents any kind of images. Videos are not included, though animated images (like animated gif) are described with an image type.	image/gif image/png image/jpeg image/bmp image/webp
audio	Represents any kind of audio files	audio/midi audio/mpeg audio/webm audio/ogg audio/wav
video	Represents any kind of video files	video/webm video/ogg
application	Represents any kind of binary data.	application/javascript application/octet-stream application/pkcs12 application/json application/xml application/pdf

Cookie Request/Response Headers



- **A cookie represents a small piece of data**

- **Server writes the Cookie to the Client (`set-cookie:…`)**
- **Client transmits all Cookies for the current site back to the Server (`cookie:…`)**

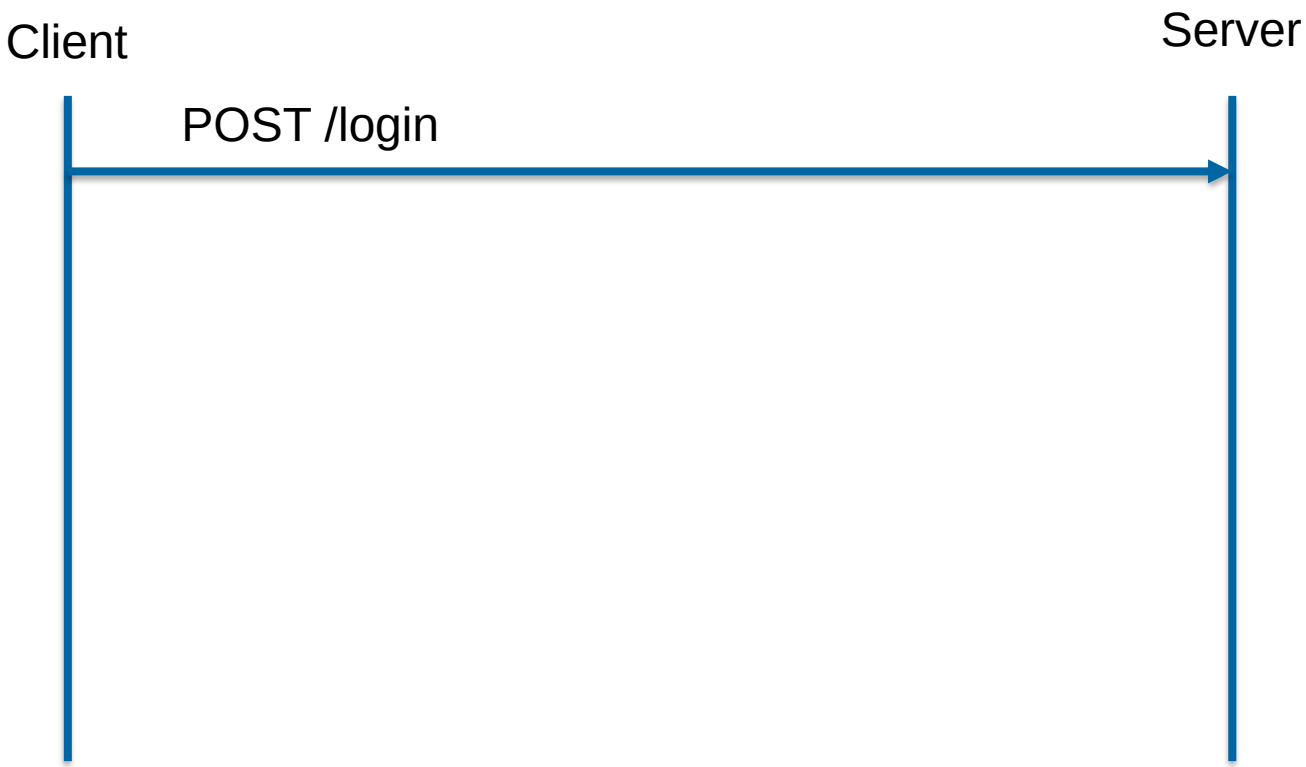
Cookie

- **Invented by Netscape as HTTP Header**
  - Originally a non-standard header
  - Today standardized in RFC 6265
- **Cookie is a small data unit stored on the Client and transmitted to Server on every Request**
  - Max 4096 Bytes
  - Max 50 Cookies per domain (varies by browser)
  - Max 3000 Cookies overall (varies by browser)
- **Cookie Expiration Time can be declared**
  - If a Cookie has no expiration, the Cookie is valid until the browser gets closed
  - Memory only Cookies are treaded as **Session**
- **Cookie can be declared as HTTPOnly to use the cookie only for HTTP / HTTPS requests**
- **Cookie can be declared as Secure to use the cookie only for HTTPS requests**

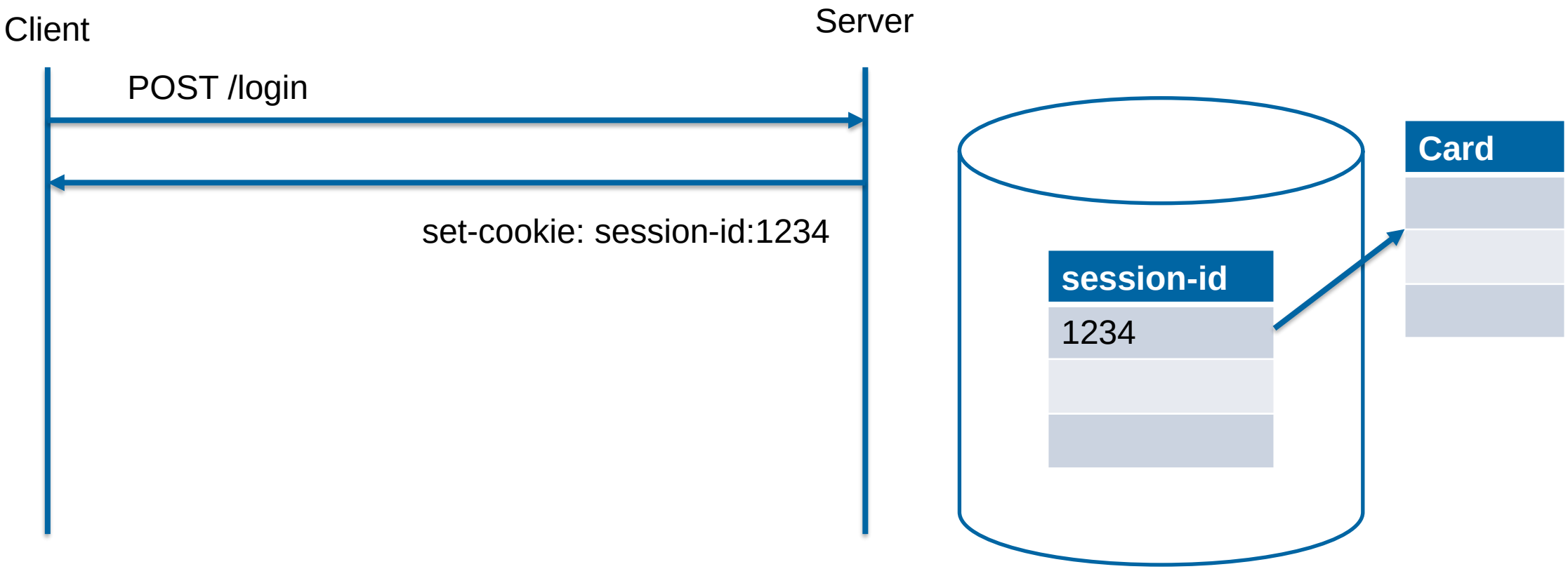
SESSION MANAGEMENT

HTTP

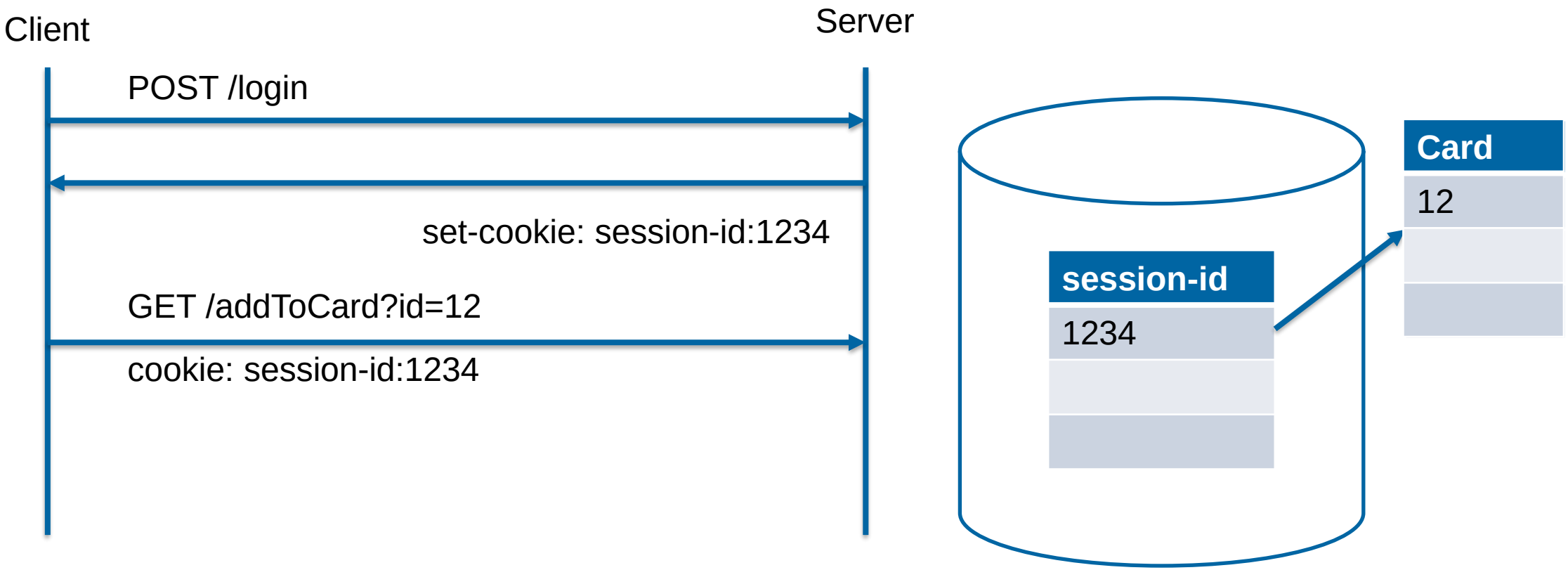
Session: Idee



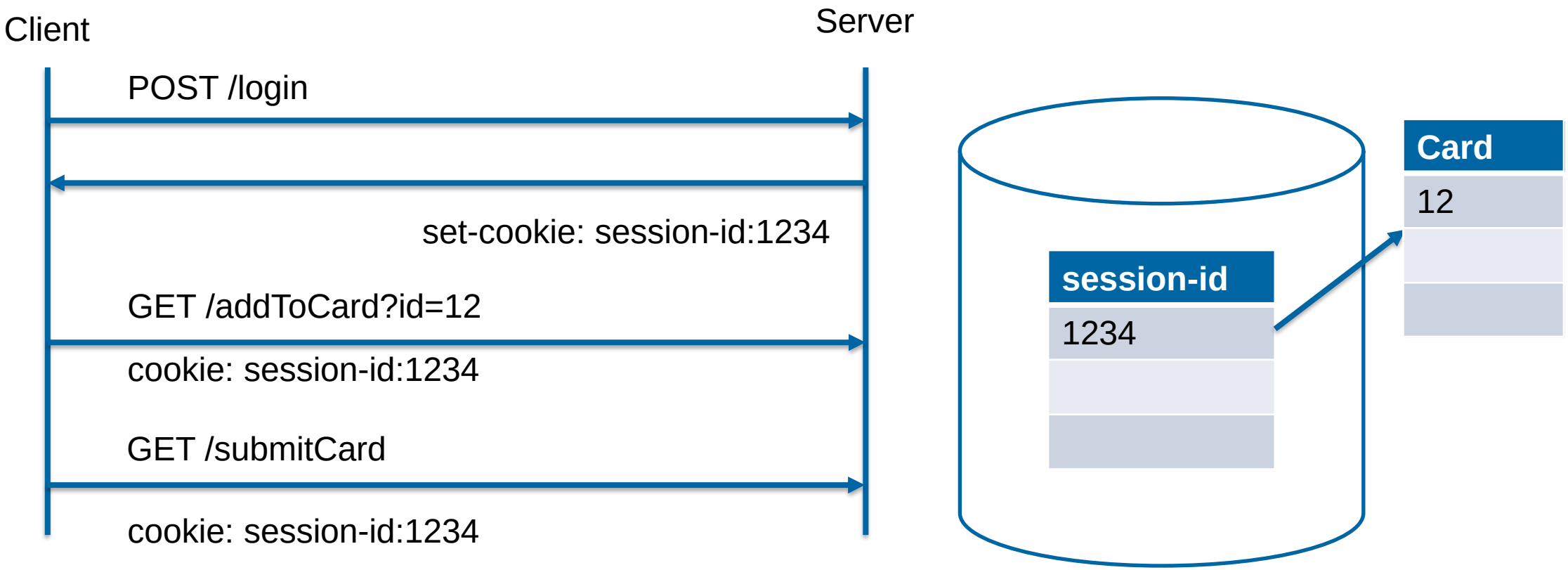
Session: Idee



Session: Idee



Session: Idee



TOOLS

Network Connectivity

- **ping** **Check Connectivity** **OSI Layer 3** **ICMP**

```
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
Reply from 192.168.1.1: bytes=32 time=1ms TTL=64
```

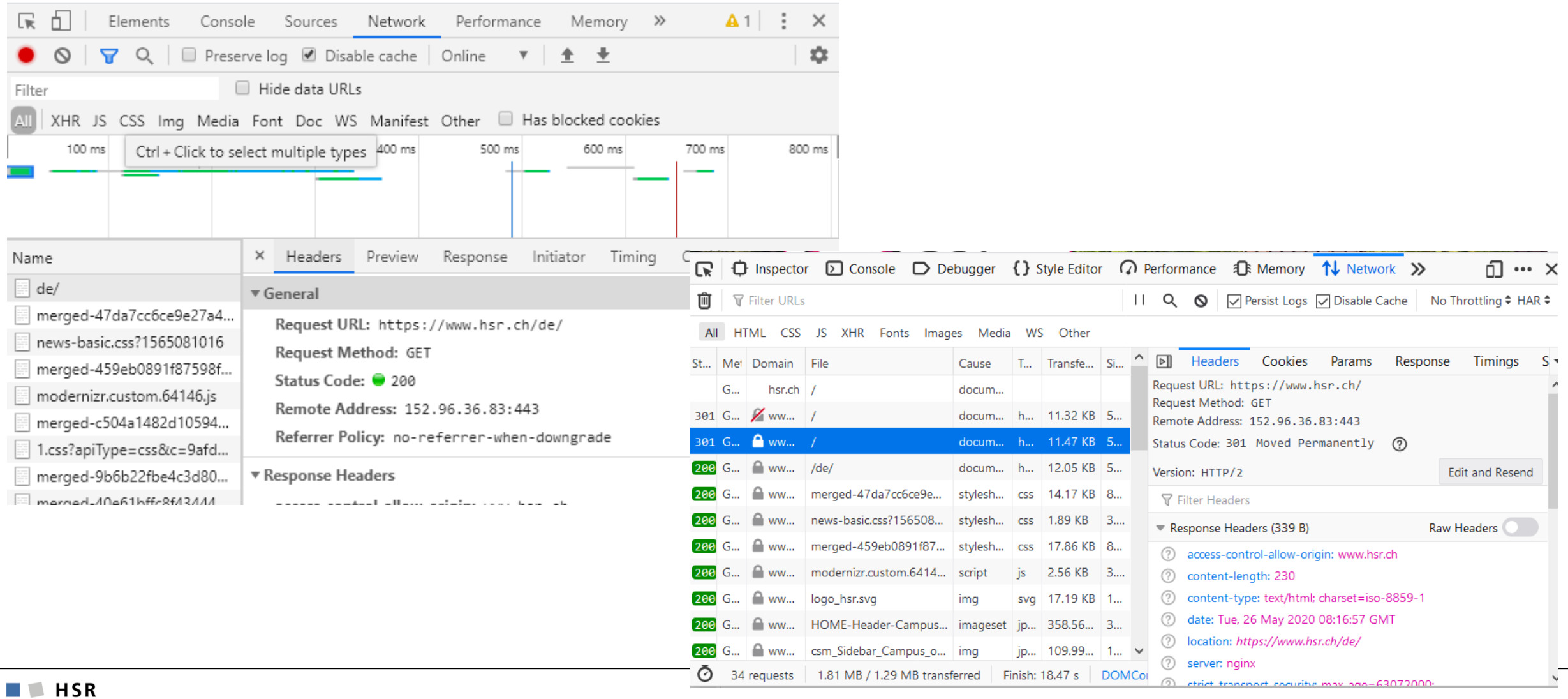
- **tracert / traceroute (Unix)** **Check IP Nodes** **OSI Layer 3** **ICMP**

```
C:\>tracert www.namics.com

Tracing route to www.namics.com [81.18.24.155]
over a maximum of 30 hops:

  1     1 ms    <1 ms    <1 ms  ZyXEL.Home [192.168.1.1]
  2    14 ms    14 ms    15 ms  161-242-3-213.bluewin.ch [213.3.242.161]
  3    14 ms    15 ms    14 ms  122-246-3-213.bluewin.ch [213.3.246.122]
  4    15 ms    15 ms    15 ms  66-0-186-195.bluewin.ch [195.186.0.66]
  5    15 ms    15 ms    15 ms  66-0-186-195.bluewin.ch [195.186.0.66]
  6    14 ms    14 ms    15 ms  i79tix-035-hun0-0-0.bb.ip-plus.net [138.187.129.
55]
  7     *        *        *     Request timed out.
  8    16 ms    15 ms    15 ms  bu-Ether1-200.lcr01.eqx004.bb.fcom.ch [212.60.63
.192]
  9    17 ms    16 ms    17 ms  ge0-1.ar01.bie005.bb.fcom.ch [213.221.198.58]
 10    21 ms    16 ms    17 ms  ge0-1.ar01.bie005.bb.fcom.ch [213.221.198.58]
 11    17 ms    17 ms    17 ms  81.18.24.155

Trace complete.
```

MacOS

```
traceroute -P icmp www.namics.com
```

Dev-Tools


Elements
Console
Sources
Network
Performance
Memory
Preserve log
Disable cache
Online
Filter
Hide data URLs
All XHR JS CSS Img Media Font Doc WS Manifest Other
Has blocked cookies
Ctrl + Click to select multiple types
100 ms
400 ms
500 ms
600 ms
700 ms
800 ms
Name
de/
merged-47da7cc6ce9e27a4...
news-basic.css?1565081016
merged-459eb0891f87598f...
modernizr.custom.64146.js
merged-c504a1482d10594...
1.css?apiType=css&c=9afd...
merged-9b6b22fbe4c3d80...
Headers
Preview
Response
Initiator
Timing
General
Request URL: https://www.hsr.ch/de/
Request Method: GET
Status Code: 200
Remote Address: 152.96.36.83:443
Referrer Policy: no-referrer-when-downgrade
Response Headers
Inspector
Console
Debugger
Style Editor
Performance
Memory
Network
Filter URLs
Persist Logs
Disable Cache
No Throttling
HAR
All HTML CSS JS XHR Fonts Images Media WS Other
St... Me Domain File Cause T... Transfe... Si...
301 G... ww... / docum... h... 11.32 KB 5...
301 G... ww... / docum... h... 11.47 KB 5...
200 G... ww... /de/ docum... h... 12.05 KB 5...
200 G... ww... merged-47da7cc6ce9e... stylesh... css 14.17 KB 8...
200 G... ww... news-basic.css?156508... stylesh... css 1.89 KB 3....
200 G... ww... merged-459eb0891f87... stylesh... css 17.86 KB 8...
200 G... ww... modernizr.custom.6414... script js 2.56 KB 3....
200 G... ww... logo_hsr.svg img svg 17.19 KB 1...
200 G... ww... HOME-Header-Campus... imageset jp... 358.56... 3...
200 G... ww... csm_Sidebar_Campus_o... img jp... 109.99... 1...
34 requests
1.81 MB / 1.29 MB transferred
Finish: 18.47 s
Headers
Cookies
Params
Response
Timings
Request URL: https://www.hsr.ch/
Request Method: GET
Remote Address: 152.96.36.83:443
Status Code: 301 Moved Permanently
Version: HTTP/2
Edit and Resend
Filter Headers
Response Headers (339 B)
Raw Headers
access-control-allow-origin: www.hsr.ch
content-length: 230
content-type: text/html; charset=iso-8859-1
date: Tue, 26 May 2020 08:16:57 GMT
location: https://www.hsr.ch/de/
server: nginx

ÜBUNGEN

Übungsserie 2 – HTTP & Session Management

Führen die folgenden Übungen mit den Developer-Tools Ihres Browsers durch **20’**

- **Öffnen Sie Google Chrome im Inkognito-Moduls und aktivieren Sie die Developer Console [F12 im Browser drücken, Network Tab anwählen]. Wählen Sie im Anschluss google.ch an.**
- **Analysieren Sie die folgenden Request/Response Headers:**
  - Mit welcher Web Method wurde google.ch angefragt?
  - Welchen Status-Code erhalten Sie für google.ch?
  - Wurde ein Session Cookie angelegt?
  - Zu welchem Zweck können Cookies noch verwendet werden?
- **Tippen Sie nun eine beliebige Suchanfrage ins Suchfeld. Analysieren Sie die `search?...` Anfrage:**
  - Welcher content-type wurde vom Server zurück gegeben?
- **Fakultativ**
  - Analysieren Sie die weiteren Headers; was gibt der Browser über sich preis?
  - Vergleichen Sie mehrere Browser/Systeme.

Tip: Bearbeiten Sie die Fragen im Team mit Ihrem Nachbarn.

QUESTIONS?

Sources

- **Slides**
  - Silvan Gehrig, HSR
  - Michael Gfeller, HSR
- **Bild-Quellen**
  - https://developer.mozilla.org
  - http://en.wikipedia.org/